# 第二章 CSS选择器

W3C在CSS3的工作草案中把选择器独立出来成为一个模块。实际上，选择器是CSS知识中的重要组成成分之一，也是CSS的根基。利用CSS选择器能不改动HTML结构，通过添加不同的CSS规则得到不同样式的网页。

2.1 认识CSS选择器

要使某个样式应用于特定的HTML元素，首先需要找到该元素。在CSS中，执行这一任务的表现规则称为CSS选择器。它为获取目标元素之后施加样式提供了极大的灵活性。实际上，CSS2.1已经为大家提供了很多常用的选择器，基本能满足Web设计师常规的设计需求。

2.1.1 CSS3选择器的优势

CSS3选择器在常规选择器的基础上新增了属性选择器、伪类选择器、过滤选择器。可以帮助您在开发过程中减少对HTML类名或ID名的依赖，以及对HTML元素的结构依赖，使编写代码更加简单轻松。

2.1.2 CSS3选择器分类

根据所获取页面中元素的不同，把CSS3选择器分为五大类：基本选择器、层次选择器、伪类选择器、伪元素和属性选择器。其中伪类选择器又分为六种：动态伪类选择器、目标伪类选择器、语言伪类选择器、UI元素状态伪类选择器、结构伪类选择器和否定伪类选择器。

2.2 基本选择器

基本选择器是CSS中使用最频繁、最基础，也是CSS中最早定义的选择器，这部分选择器在CSS1中就定义了，为了便于初学者温故知新，不妨回顾CSS的基础选择器。

2.2.1 基本选择器语法

通过基本选择器可以确定HTML树形结构中大多数的DOM元素节点。

|  |  |  |
| --- | --- | --- |
| 选择器 | 类型 | 功能描述 |
| \* | 通配选择器 | 选择文档中所有的HTML元素 |
| E | 元素选择器 | 选择指定的类型的HTML元素 |
| #id | ID选择器 | 选择指定ID属性为“id”的任意类型元素 |
| .class | 类选择器 | 选择指定class属性值为“class”的任意类型的任意多个元素 |
| Selector1，selectorN | 群组选择器 | 将每一个选择器匹配的元素集合并 |

2.2.2 浏览器兼容性

浏览器对基本选择器都是一路绿灯通行。

2.2.3 实战体验：使用基本选择器

下面通过示例介绍各种基本选择器在页面中的使用方法。

页面中有一个列表，其中第一个和最后一个设置了ID属性，其中各部分列表项设置了class类名，通过基本选择器来改变元素的样式风格。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>使用CSS3基本选择器</title>

<style type="text/css">

\*{

margin: 0;padding: 0;

}

.clearfix:after,.clearfix:before{display: table;content:""}

.clearfix:after{clear:both;overflow: hidden;}

.demo{ width:250px;border:1px solid #ccc;padding:10px;margin: 20px auto;}

li{

list-style:none outside none;

float: left;

height:20px;

line-height: 20px;

width:20px;

border-radius: 10px;

text-align: center;

background-color: #f36;

color: green;

margin-right: 5px;

}

</style>

</head>

<body>

<ul class="clearfix demo">

<li class="first" id="first">1</li>

<li class="active">2</li>

<li class="important item">3</li>

<li class="important">4</li>

<li class="item">5</li>

<li>6</li>

<li>7</li>

<li>8</li>

<li>9</li>

<li class="last" id="last">10</li>

</ul>

</body>

</html>

上面代码使用了基本选择器，首先看看页面的初步效果。下面通过图解的方法说明CSS3基本选择器的使用方法。

2.2.4 通配选择器

通配选择器（\*）用来选择所有元素，当然也可以选择某个元素下的所有元素。如：

\*{ margin：0； padding：0 }

上面一行代码大家在Reset样式文件中经常看到，表示所有元素的margin和padding都设置为0。为了更好地说明问题，通过CSS3选择器中的通配选择器来改变列表中的所有子项的背景色设置为orange。

.demo \* { background:orange}

此时元素类名为demo下的所有元素都将背景色设置为橙色。

2.2.5 元素选择器

元素选择器（E）是CSS选择器中最常见、最基本的选择器。文档的元素包括html、body、p、div 等，如示例中ul 、li 也属于html 元素 。接下来通过ul元素选择器改变整个列表的背景色。

ul { background:grey; }

2.2.6 ID选择器

在设置ID选择器之前，需要在html文档中给对应的元素设置id属性并设置其值。在CSS样式中使用id选择器时，需要在id属性值得上面加上“#”号。在下面这个示例中，可以轻松地看到列表的第一项和最后一项分别定义了一个id，其值分别为“first”和“last”。使用这两个id值来改变列表项中第一个和最后一个列表项的背景色和前景色，代码如下。

#first{background:lime;color:#000}

#last{background:#000;color:lime}

2.2.7 类选择器

类选择器（.class）是以独立于文档元素的方式来指定元素样式。使用方法与ID选择器极其相似，首先在HTML给需要的元素定义class属性，并为其设置属性值。其与ID选择器有一个很大不同之处。“类选择器在页面中可以有多个相同的类名，而ID选择器其ID值在整个页面中是唯一的一个”。同样，看看如何通过类选择器来改变元素的样式。

.item{background:green;color:#fff;font-weight:bold}

上面是类选择器的简单使用，其实类选择器还有一种使用的方法，就是多类选择器。通过两个或两个以上类选择器合并，来定义有别于一个类名的元素效果。

.item.important{background:red}

使用多类选择器时，大家需要注意，如果一个多类选择器包含的类名中其中一个不存在，这个选择器将无法找到相匹配的元素，正如上面的代码，其只能匹配li元素同时具有“item”

和“important”的元素，而只有其中任何一个类名都将无法匹配。

由于类名在一个HTML文档中可以同时存在于不同的元素标签上。换句话说，在一个HTML文档中，div可以有类名“block”，ul也可以有类名“block”，但有时在web的页面开发中，仅需要对ul为“block”定义样式，此时仅采用类名选择器并不能达到需要的效果，其CSS选择器还支持带有标签的类名选择器“ul.block”。

ul.block{background:#CCC;}

上面代码中只匹配class属性包含“block”的所有ul元素，但其他任何类型的元素都不匹配，包括有“block”类名的元素。简而言之，“ul.block”只会匹配ul元素，并且有一个类名“block”。不符合这两个条件的任何一个都不能与选择器匹配。

2.2.8 群组选择器

群组选择器是将具有相同样式的元素分组到一起，每个选择器之间用逗号隔开，例如“selector1，selector2，···，selectorN”。这个逗号告诉浏览器，规则中包含多个不同的选择器，省去逗号就成了后代选择器，这一点大家在使用中千万小心。

2.3 层次选择器

层次选择器通过HTML的DOM元素间的层次关系获取元素，其主要的层次关系包括后代、父子、相邻兄弟和通用兄弟几种关系，通过其中某类关系可以方便快捷地选定需要的元素。

2.3.1 层次选择器的语法

层次选择器是一个非常好的选择器，也是大家常用的选择器。

|  |  |  |
| --- | --- | --- |
| 选择器 | 类型 | 功能描述 |
| E F | 后代选择器 | 选择匹配的F元素，且匹配的F被包含在匹配的E元素内 |
| E > F | 子选择器 | 选择匹配的F元素，且匹配的F元素是所匹配的E元素的子元素 |
| E+F | 相邻兄弟选择器 | 选择匹配的F元素，且匹配的F元素仅位于匹配的E元素后边 |
| E~F | 通用兄弟选择器 | 选择匹配的F元素，且位于匹配的E元素后的所有匹配的F元素 |

2.3.3 浏览器兼容性

子选择器、相邻兄弟选择器和通用选择器要IE7以及其以上版本才支持。

2.3.3 实战体验：使用层次选择器选择元素

在层次选择器中，后代选择器与子选择器是比较常用的，而对于相邻兄弟选择器和通用兄弟选择器而言，大家平常并不常使用，特别是CSS3选择器中新增的通用兄弟选择器。下面通过示例来演示各种层次选择器在页面中如何选择HTML的DOM元素的方法。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>使用CSS3层次选择器</title>

<style>

\*{margin: 0;padding: 0;}

body{width:300px;margin: 0 auto;}

div{

margin: 5px;

padding: 5px;

border:1px solid #ccc;

}

</style>

</head>

<body>

<div>1</div>

<div>2</div>

<div>3</div>

<div>4

<div>5</div>

<div>6</div>

</div>

<div>7

<div>8

<div>9

<div>10</div>

</div>

</div>

</div>

</body>

</html>

2.3.4 后代选择器

后代选择器（E F）也称为包含选择器，作用就是可以选择某元素的后代元素。例如“E F”，E为祖先元素，F为后代元素，表达的意思就是选择E元素的所有后代F元素。这里的F元素不管是E元素的子元素、孙辈元素或者是更深层次的关系，都将被选中。换句话说，不论F在E中有多少层关系，F元素都将被选中。接下来使用后代选择器改变其背景色。

div div{background:orange;}

2.3.5 子选择器

子选择器（E > F）只能选择某元素的子元素，其中E为父元素，而F为子元素，其中E>F 表示选择了E元素下所有子元素F。这与后代选择器（E F）不一样，在后代选择器中F是E的后代元素，而在E>F中F仅仅是E的子元素而已。接下来的例子，选择器改变body下的子元素div的背景色。

body>div{background:green;}

2.3.6 相邻兄弟选择器

相邻兄弟选择器（E + F）可以选择紧接在另一个元素后的元素，它们具有一个相同的父元素。换句话说，E和F是同辈元素，F元素在E元素后边，而且相邻，这样就可以使用相邻兄弟元素选择器来选择F元素。

.active +div {background:lime;}

2.3.7 通用兄弟选择器

通用兄弟选择器（E ~ F）是CSS3新增加的，用于选择某元素后面的所有兄弟元素，它们和相邻兄弟选择器类似，需要在同一个父元素之中。也就是说，E和F元素都是同辈元素，并且F元素在E元素之后，E~F元素将选中E元素后面所有F元素。

.active ~div{background:red}

2.4 动态伪类选择器

伪类选择器对于大家来说最熟悉的莫过于“:link”、“:visited”、“:hover”、“:avtive”，因为这些是大家平时常用到的伪类选择器。而CSS3的伪类选择器可以分为六种：动态伪类选择器、目标伪类选择器、语言伪类选择器、UI状态伪类选择器、结构伪类选择器和否定伪类选择器。

伪类选择器语法书写时和其他的选择器写法有所区别，都以冒号开头。例如：

E:pseudo-class{property:value;}

其中E为HTML中的元素；pseudo-class是CSS的伪类选择器名称；property是CSS的属性；value为CSS属性值。

CSS3伪类选择器有什么功能？选定元素能带来什么便利？带着这些问题，依次学习CSS3伪类选择器的使用方法，首先是动态伪类选择器。

2.4.1 动态伪类选择器语法

动态伪类早在CSS中就有，并不是CSS3才有的，动态伪类并不存在HTML中，只有当用户和网站交互的时候才能体现出来。第一种是在链接中常看到的锚点伪类，另一种为用户行为伪类。

|  |  |  |
| --- | --- | --- |
| 选择器 | 类型 | 功能描述 |
| E:link | 链接伪类选择器 | 选择匹配的E元素，而且匹配元素被定义了超链接并未被访问过。常用于链接锚点上。 |
| E:visited | 链接伪类选择器 | 选择匹配的E元素，并且匹配元素被定义了超链接并已被访问过。 |
| E:active | 用户行为伪类选择器 | 选择匹配的E元素，且匹配元素被激活。常用于锚点和按钮上 |
| E:hover | 用户行为伪类选择器 | 选择匹配的E元素，且用户鼠标在停留在元素E上。 |
| E:focus | 用户行为伪类选择器 | 选择匹配的E元素，且匹配的元素获得焦点 |

2.4.3 实战体验：美化按钮

在众多网站上按钮在不同状态下效果不一，用以增强用户体验，这也是一种非常好的设计体验与细节。实现并不复杂。

根据用户的行为不同，按钮效果可以分为：默认状态、悬浮状态、点击时状态、焦点状态和点击后状态，可以按照CSS3的动态伪类选择器，在不同状态下给按钮赋予不同的样式风格。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>使用动态伪类选择器美化按钮</title>

<style>

.download-info{

text-align: center;

}

/\*默认状态下的按钮效果\*/

.bn{

background-color: #0074CC;

\*background-color: #0055CC;

/\*CSS3渐变制作背景图片\*/

background-image: -ms-linear-gradient(top,#0088CC,#0055cc);

background-image: -webkit-gradient(linear,0 0, 0 100%,from(#0088CC),to("#0055CC"));

background-image: -webkit-linear-gradient(top,#0088CC,#0055cc);

background-image: -moz-linear-gradient(top,#0088CC,#0055cc);

background-image: linear-gradient(top,#0088cc,#0055cc);

display: inline-block;

\*display: inline;

border: 1px solid #cccccc;

\*border:0;

border-color: #ccc;

/\*CSS3的色彩模块\*/

border-color: rgba(0,0,0,0.1) rgba(0,0,0,0.1) rgba(0,0,0,0.25);

border-radius: 6px;

color: #ffffff;

cursor: pointer;

font-size: 20px;

font-weight: normal;

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr="#0088cc",endColorstr="#0055cc",GradientType=0);

filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);

line-height: normal;

padding: 14px 24px;

text-align: center;

text-shadow: 0 -1px 0 rgba(0,0,0,0.25);

text-decoration: none;

vertical-align: middle;

\*zoom: 1;

}

.bn:hover{

background-position: 0 -15px;

background-color: #0055cc;

\*background-color:#004ab3;

color:#FFFFFF;

text-decoration: none;

text-shadow: 0 -1px 0 rgba(0,0,0,0.25);

/\*CSS3动画效果\*/

-webkit-transition: background-position 0.1s linear;

-moz-transition: background-position 0.1s linear;

-ms-transition: background-position 0.1s linear;

transition: background-position 0.1s linear;

}

/\*点击时按钮效果\*/

.bn:active{

background-color: #0055CC;

background-image: none;

outline: 0;

box-shadow: inset 0 2px 4px rgba(0,0,0,0.15),0 1px 2px rgba(0,0,0,0.05);

color: rgba(255,255,255,0.75);

}

.bn:focus{

outline: thin dotted #333;

outline: 5px auto -webkit-focus-ring-color;

outline-offset:-2px ;

}

</style>

</head>

<body>

<div class="download-info">

<a href="#" class="bn">view project on GitHub</a>

</div>

</body>

</html>

2.5 目标伪类选择器

目标伪类选择器“:target”是众多的CSS3特性中的一个，用来匹配文档（页面）的URI某个标识符的目标元素。具体来说，URI中的标识符通常会包含一个井号（#），后面带有一个标识符名称，例如“#contact”“:target”就是用来匹配ID为“contact”的元素的。换中说法，在web页面中，一些URL拥有片段标识符，它是由一个井号（#）后跟一个锚点或元素ID组合而成，可以链接到页面的某个特定元素。“:target”伪类选择器选取链接的目标元素，然后供定义样式。

2.5.3 实战体验 制作手风琴效果

以前制作手风琴效果需要依赖JavaScript脚本。CSS3的目标伪类选择器可以不使用任何JavaScript代码实现手风琴效果。

页面中有三个区块，默认状态只显示三个区块的标题，点击其中一个标题时，其对应的内容就会显示；点击另一个标题时，对应区块内容将显示，而前一块内容将隐藏。

通过目标伪类选择器“E:target”,显示和隐藏不同栏目的内容，从而实现手风琴效果。

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>垂直手风琴</title>

<style>

.accordionMenu{

background: #fff;

color: #424242;

font:12px arial,verdana,sans-serif;

margin: 0 auto;

padding: 10px;

width:500px;

}

.accordionMenu h2{

margin: 5px 0;

padding: 0;

position: relative;

}

.accordionMenu h2:before{

border: 5px solid #FFF;

border-color: #fff transparent transparent;

content: "";

height: 0;

position: absolute;

right: 10px;

top: 15px;

width: 0;

}

.accordionMenu h2 a{

background: #8f8f8f;

background: linear-gradient(top,#cecece,#8f8f8f);

border-radius: 5px;

color: #424242;

display: block;

font-size: 13px;

font-weight: normal;

margin: 0;

padding: 10px 10px;

text-shadow: 2px 2px 2px #aeaeae;

text-decoration: none;

}

.accordionMenu :target h2 a,

.accordionMenu h2 a:focus,

.accordionMenu h2 a:hover,

.accordionMenu h2 a:active{

background: #2828dd;

background: linear-gradient(top,#6bb2ff,#2288dd);

color: #FFF;

}

.accordionMenu p{

margin: 0;

height: 0;

overflow: hidden;

padding: 0 10px;

transition: height 0.5s ease-in;

}

.accordionMenu :target p{

height: 100px;

overflow: auto;

}

.accordionMenu :target h2:before{

border-color:transparent transparent transparent #fff;

}

</style>

</head>

<body>

<div class="accordionMenu">

<div class="menuSection" id="brand">

<h2><a href="#brand">Brand</a></h2>

<p>Lorem ispsum dolor...</p>

</div>

<div class="menuSection" id="promotion">

<h2><a href="#promotion">Prometion</a></h2>

<p>Lorem issum dolor sit amet...</p>

</div>

<div class="menuSection" id="event">

<h2><a href="#event">Event</a></h2>

<p>Lorem issum dolor sit amet...</p>

</div>

</div>

</body>

</html>

2.6 语言伪类选择器

使用语言伪类选择器来匹配使用语言的元素是非常有用的，特别是用于多语言版本的网站，其作用更是明显。可以使用他来根据不同语言版本设置页面的字体风格。

2.6.1 语言伪类选择器语法

语言伪类选择器是根据元素的语言编码匹配元素。这种语言信息必须包含在文档中，或者与文档关联，不能从CSS指定。为文档指定语言有两种方法可以表示。如果使用HTML5，直接可以设置文档的语言。例如：

<!DOCTYPE HTML>

<html lang=”en-US”>

另一种方法就是手工在文档中指定lang属性，并设置对应的语言值。例如：

<body lang=”fr”>

语言伪类选择器允许为不同的语言定义特殊的规则，这在多语言版本的网站用起来是特别的方便。

E:lang（language）表示选择匹配E的所有元素，且匹配元素指定了lang属性，且其值为language。

2.7 UI元素状态伪类选择器

UI元素状态伪类选择器也是CSS3选择器模块组中的一部分，主要用于form表单元素上，以提高网页的人机交互、操作逻辑以及页面的整体美观，使表单页面更具个性与品位，而且使用户操作页面表单更便利和简单。

2.7.1 UI元素状态伪类选择器语法

UI元素的状态一般包括：启用、禁用、选中、未选中、获得焦点、失去焦点、锁定和待机等。在HTML元素中有可用和不可用状态，例如表单中的文本输入框；HTML元素中还有选中和未选中状态，例如表单中的复选按钮和单选按钮。这几种状态都是CSS3选择器中常用的状态伪类选择器。

|  |  |  |
| --- | --- | --- |
| 选择器 | 类型 | 功能描述 |
| E:checked | 选中状态伪类选择器 | 匹配选中的复选按钮或单选按钮表单元素 |
| E:enabled | 启用状态伪类选择器 | 匹配所有启用的表单元素 |
| E:disabled | 不可用状态伪类选择器 | 匹配所有禁用的表单元素 |

2.8 结构伪类选择器

伪类可以将一段并不存在的HTML当作独立元素来定位，或是找到无法使用其他简单选择器就能定位的切实存在的元素。因此CSS3给伪类选择器引入一种“结构伪类选择器”。这种选择器可以根据元素在文档中的某些特性定位到他们，通过文档树结构的相互关系来匹配特定的元素，从而减少HTML文档对ID或类名的定义，帮助你保持代码干净和整洁。

2.8.1 重温HTML的DOM树

所有的结构伪类都是基于HTML文档树的，也称做文档对象模型（DOM），下面简单回顾一下这方面的知识。文档树是HTML页面的层级结构。它由元素、属性和文本组成，它们都是一个节点，就像公司的结构组织图一样。

2.8.2 结构伪类选择器语法

|  |  |
| --- | --- |
| 选择器 | 功能描述 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |